Case Study 1

Suvradri Maitra

25/07/2022

Contents

Executive Summary

Cyclistic is a successful bike-sharing offering. It provides services in the city of Chicago, IL. The bikes can be unlocked from one station and returned to any other station in the system anytime.

The aim of this analysis is to establish the way in which “members”, who purchase annual memberships, and “casual riders”, who purchase singe-ride or full-day passes, use Cyclistic bikes differently.

The company’s success is due to its flexibility of the pricing plans - single-ride passes, full-day passes, and annual memberships.

Ask Phase

Business Task

Analyzing the Cyclistic historical bike trip data to identify trends and understand how annual members and casual riders differ.

Stakeholders

- Lily Moreno, director of marketing and my manager
- Cyclistic marketing analytics team
- Cyclistic executive team

Context

The finance analysts have noted that annual members are much more profitable than the casual riders. Even though pricing flexibility attracts more customers to this bike-sharing platform, maximizing the number of annual members is the key to future growth.

My analysis is the first of the three steps to design marketing strategies aimed at converting casual riders into annual members.

Prepare Phase

Dataset Used

The data source used for our case study is previous 12 months of Cyclistics trip data made available by Motivate International Inc. under this license.

Accessibility and Privacy

This is public data that is used to explore how different customer types are using Cyclistic bikes. Usage of personally identifiable information is prohibited.

Data Organization

Available to us are different archived folders, or compressed, or "zipped" files described as follows - monthly data from April 2020 to June 2022 - quarterly data of 2014, 2015, 2016, 2017, 2018, 2019 and Q1 of 2020 - yearly data of 2013

Each of the compressed folders have CSV files. The CSV documents are extracted and stored locally for the purpose of this analysis. Each document represents the quantitative data tracked by Cyclistic. The data is considered wide, since each row is an observation of each ride, so each ride has information in multiple columns. Every ride has a unique ID.

Data Security

The data set has been carefully stored during the time period of this analysis and deleted after the case study has been completed to ensure data security.

Data Verification

Since the data is large, I cannot use spreadsheet tools like Microsoft Excel or Google Sheets for the work, and chose to do all activities using RStudio Desktop Application.

Imported last 12 months data to the RStudio Environment to verify the files

```
#loading packages required for the work
library(readr)
library(ggplot2)
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.2 --
## v tibble  3.1.7     v dplyr   1.0.9
## v tidyr   1.2.0     v stringr 1.4.0
## v purrr   0.3.4     v forcats 0.5.1
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(lubridate)
```

```
##
## Attaching package: 'lubridate'
##
## The following objects are masked from 'package:base':
##
##     date, intersect, setdiff, union
```

```
library(dplyr)
library(skimr)
library(janitor)
```

```
##
## Attaching package: 'janitor'
```

```
##
## The following objects are masked from 'package:stats':
##
##     chisq.test, fisher.test
```

```
#importing last 12 months data
jul21 <- read.csv("202107.csv")
aug21 <- read.csv("202108.csv")
sep21 <- read.csv("202109.csv")
oct21 <- read.csv("202110.csv")
nov21 <- read.csv("202111.csv")
dec21 <- read.csv("202112.csv")
jan22 <- read.csv("202201.csv")
feb22 <- read.csv("202202.csv")
mar22 <- read.csv("202203.csv")
apr22 <- read.csv("202204.csv")
may22 <- read.csv("202205.csv")
jun22 <- read.csv("202206.csv")
```

Getting a preview of the files

```
head(jul21, n = 5)
```

```
##            ride_id rideable_type          started_at            ended_at
## 1 0A1B623926EF4E16   docked_bike 2021-07-02 14:44:36 2021-07-02 15:19:58
## 2 B2D5583A5A5E76EE  classic_bike 2021-07-07 16:57:42 2021-07-07 17:16:09
## 3 6F264597DDBF427A  classic_bike 2021-07-25 11:30:55 2021-07-25 11:48:45
## 4 379B58EAB20E8AA5  classic_bike 2021-07-08 22:08:30 2021-07-08 22:23:32
## 5 6615C1E4EB08E8FB electric_bike 2021-07-28 16:08:06 2021-07-28 16:27:09
##              start_station_name start_station_id              end_station_name
## 1 Michigan Ave & Washington St            13001   Halsted St & North Branch St
## 2   California Ave & Cortez St            17660           Wood St & Hubbard St
## 3         Wabash Ave & 16th St           SL-012           Rush St & Hubbard St
## 4   California Ave & Cortez St            17660        Carpenter St & Huron St
## 5   California Ave & Cortez St            17660 Elizabeth (May) St & Fulton St
##   end_station_id start_lat start_lng  end_lat   end_lng member_casual
## 1   KA1504000117  41.88398 -87.62468 41.89937 -87.64848        casual
## 2          13432  41.90036 -87.69670 41.88990 -87.67147        casual
## 3   KA1503000044  41.86038 -87.62581 41.89017 -87.62619        member
## 4          13196  41.90036 -87.69670 41.89456 -87.65345        member
## 5          13197  41.90035 -87.69668 41.88659 -87.65839        casual
```

```
head(aug21, n = 5)
```

```
##            ride_id rideable_type          started_at            ended_at
## 1 99103BB87CC6C1BB electric_bike 2021-08-10 17:15:49 2021-08-10 17:22:44
## 2 EAFCCCFB0A3FC5A1 electric_bike 2021-08-10 17:23:14 2021-08-10 17:39:24
## 3 9EF4F46C57AD234D electric_bike 2021-08-21 02:34:23 2021-08-21 02:50:36
## 4 5834D3208BFAF1DA electric_bike 2021-08-21 06:52:55 2021-08-21 07:08:13
## 5 CD825CB87ED1D096 electric_bike 2021-08-19 11:55:29 2021-08-19 12:04:11
##   start_station_name start_station_id end_station_name end_station_id start_lat
## 1                                                                        41.77
```

```
## 2                                                                    41.77
## 3                                                                    41.95
## 4                                                                    41.97
## 5                                                                    41.79
##   start_lng end_lat end_lng member_casual
## 1    -87.68   41.77  -87.68        member
## 2    -87.68   41.77  -87.63        member
## 3    -87.65   41.97  -87.66        member
## 4    -87.67   41.95  -87.65        member
## 5    -87.60   41.77  -87.62        member
```

```
head(sep21, n = 5)
```

```
##            ride_id rideable_type          started_at            ended_at
## 1 9DC7B962304CBFD8 electric_bike 2021-09-28 16:07:10 2021-09-28 16:09:54
## 2 F930E2C6872D6B32 electric_bike 2021-09-28 14:24:51 2021-09-28 14:40:05
## 3 6EF72137900BB910 electric_bike 2021-09-28 00:20:16 2021-09-28 00:23:57
## 4 78D1DE133B3DBF55 electric_bike 2021-09-28 14:51:17 2021-09-28 15:00:06
## 5 E03D4ACDCAEF6E00 electric_bike 2021-09-28 09:53:12 2021-09-28 10:03:44
##   start_station_name start_station_id end_station_name end_station_id start_lat
## 1                                                                    41.89
## 2                                                                    41.94
## 3                                                                    41.81
## 4                                                                    41.80
## 5                                                                    41.88
##   start_lng end_lat end_lng member_casual
## 1    -87.68   41.89  -87.67        casual
## 2    -87.64   41.98  -87.67        casual
## 3    -87.72   41.80  -87.72        casual
## 4    -87.72   41.81  -87.72        casual
## 5    -87.74   41.88  -87.71        casual
```

```
head(oct21, n = 5)
```

```
##            ride_id rideable_type          started_at            ended_at
## 1 620BC6107255BF4C electric_bike 2021-10-22 12:46:42 2021-10-22 12:49:50
## 2 4471C70731AB2E45 electric_bike 2021-10-21 09:12:37 2021-10-21 09:14:14
## 3 26CA69D43D15EE14 electric_bike 2021-10-16 16:28:39 2021-10-16 16:36:26
## 4 362947F0437E1514 electric_bike 2021-10-16 16:17:48 2021-10-16 16:19:03
## 5 BB731DE2F2EC51C5 electric_bike 2021-10-20 23:17:54 2021-10-20 23:26:10
##         start_station_name start_station_id end_station_name end_station_id
## 1 Kingsbury St & Kinzie St     KA1503000043
## 2
## 3
## 4
## 5
##   start_lat start_lng end_lat end_lng member_casual
## 1  41.88919  -87.6385   41.89  -87.63        member
## 2  41.93000  -87.7000   41.93  -87.71        member
## 3  41.92000  -87.7000   41.94  -87.72        member
## 4  41.92000  -87.6900   41.92  -87.69        member
## 5  41.89000  -87.7100   41.89  -87.69        member
```

```r
head(nov21, n = 5)
```

```
##            ride_id rideable_type          started_at            ended_at
## 1 7C00A93E10556E47 electric_bike 2021-11-27 13:27:38 2021-11-27 13:46:38
## 2 90854840DFD508BA electric_bike 2021-11-27 13:38:25 2021-11-27 13:56:10
## 3 0A7D10CDD144061C electric_bike 2021-11-26 22:03:34 2021-11-26 22:05:56
## 4 2F3BE33085BCFF02 electric_bike 2021-11-27 09:56:49 2021-11-27 10:01:50
## 5 D67B4781A19928D4 electric_bike 2021-11-26 19:09:28 2021-11-26 19:30:41
##   start_station_name start_station_id end_station_name end_station_id start_lat
## 1                                                                         41.93
## 2                                                                         41.96
## 3                                                                         41.96
## 4                                                                         41.94
## 5                                                                         41.90
##   start_lng end_lat end_lng member_casual
## 1    -87.72   41.96  -87.73        casual
## 2    -87.70   41.92  -87.70        casual
## 3    -87.70   41.96  -87.70        casual
## 4    -87.79   41.93  -87.79        casual
## 5    -87.63   41.88  -87.62        casual
```

```r
head(dec21, n = 5)
```

```
##            ride_id rideable_type          started_at            ended_at
## 1 46F8167220E4431F electric_bike 2021-12-07 15:06:07 2021-12-07 15:13:42
## 2 73A77762838B32FD electric_bike 2021-12-11 03:43:29 2021-12-11 04:10:23
## 3 4CF42452054F59C5 electric_bike 2021-12-15 23:10:28 2021-12-15 23:23:14
## 4 3278BA87BF698339  classic_bike 2021-12-26 16:16:10 2021-12-26 16:30:53
## 5 6FF54232576A3B73 electric_bike 2021-12-30 11:31:05 2021-12-30 11:51:21
##              start_station_name start_station_id           end_station_name
## 1     Laflin St & Cullerton St            13307        Morgan St & Polk St
## 2        LaSalle Dr & Huron St     KP1705001026 Clarendon Ave & Leland Ave
## 3 Halsted St & North Branch St     KA1504000117       Broadway & Barry Ave
## 4 Halsted St & North Branch St     KA1504000117      LaSalle Dr & Huron St
## 5     Leavitt St & Chicago Ave            18058     Clark St & Drummond Pl
##   end_station_id start_lat start_lng  end_lat   end_lng member_casual
## 1   TA1307000130  41.85483 -87.66366 41.87197 -87.65097        member
## 2   TA1307000119  41.89441 -87.63233 41.96797 -87.65000        casual
## 3          13137  41.89936 -87.64852 41.93758 -87.64410        member
## 4   KP1705001026  41.89939 -87.64854 41.89488 -87.63233        member
## 5   TA1307000142  41.89558 -87.68202 41.93125 -87.64434        member
```

```r
head(jan22, n = 5)
```

```
##            ride_id rideable_type          started_at            ended_at
## 1 C2F7DD78E82EC875 electric_bike 2022-01-13 11:59:47 2022-01-13 12:02:44
## 2 A6CF8980A652D272 electric_bike 2022-01-10 08:41:56 2022-01-10 08:46:17
## 3 BD0F91DFF741C66D  classic_bike 2022-01-25 04:53:40 2022-01-25 04:58:01
## 4 CBB80ED419105406  classic_bike 2022-01-04 00:18:04 2022-01-04 00:33:00
## 5 DDC963BFDDA51EEA  classic_bike 2022-01-20 01:31:10 2022-01-20 01:37:12
##               start_station_name start_station_id             end_station_name
## 1      Glenwood Ave & Touhy Ave              525          Clark St & Touhy Ave
```

```
## 2      Glenwood Ave & Touhy Ave              525          Clark St & Touhy Ave
## 3 Sheffield Ave & Fullerton Ave     TA1306000016 Greenview Ave & Fullerton Ave
## 4      Clark St & Bryn Mawr Ave     KA1504000151     Paulina St & Montrose Ave
## 5   Michigan Ave & Jackson Blvd     TA1309000002        State St & Randolph St
##   end_station_id start_lat start_lng  end_lat   end_lng member_casual
## 1         RP-007  42.01280 -87.66591 42.01256 -87.67437        casual
## 2         RP-007  42.01276 -87.66597 42.01256 -87.67437        casual
## 3   TA1307000001  41.92560 -87.65371 41.92533 -87.66580        member
## 4   TA1309000021  41.98359 -87.66915 41.96151 -87.67139        casual
## 5   TA1305000029  41.87785 -87.62408 41.88462 -87.62783        member
```

```
head(feb22, n = 5)
```

```
##            ride_id rideable_type          started_at            ended_at
## 1 E1E065E7ED285C02  classic_bike 2022-02-19 18:08:41 2022-02-19 18:23:56
## 2 1602DCDC5B30FFE3  classic_bike 2022-02-20 17:41:30 2022-02-20 17:45:56
## 3 BE7DD2AF4B55C4AF  classic_bike 2022-02-25 18:55:56 2022-02-25 19:09:34
## 4 A1789BDF844412BE  classic_bike 2022-02-14 11:57:03 2022-02-14 12:04:00
## 5 07DE78092C62F7B3  classic_bike 2022-02-16 05:36:06 2022-02-16 05:39:00
##             start_station_name start_station_id               end_station_name
## 1       State St & Randolph St     TA1305000029         Clark St & Lincoln Ave
## 2  Halsted St & Wrightwood Ave     TA1309000061 Southport Ave & Wrightwood Ave
## 3       State St & Randolph St     TA1305000029            Canal St & Adams St
## 4 Southport Ave & Waveland Ave            13235         Broadway & Sheridan Rd
## 5       State St & Randolph St     TA1305000029          Franklin St & Lake St
##   end_station_id start_lat start_lng  end_lat   end_lng member_casual
## 1          13179  41.88462 -87.62783 41.91569 -87.63460        member
## 2   TA1307000113  41.92914 -87.64908 41.92877 -87.66391        member
## 3          13011  41.88462 -87.62783 41.87926 -87.63990        member
## 4          13323  41.94815 -87.66394 41.95283 -87.64999        member
## 5   TA1307000111  41.88462 -87.62783 41.88584 -87.63550        member
```

```
head(mar22, n = 5)
```

```
##            ride_id rideable_type          started_at            ended_at
## 1 47EC0A7F82E65D52  classic_bike 2022-03-21 13:45:01 2022-03-21 13:51:18
## 2 8494861979B0F477 electric_bike 2022-03-16 09:37:16 2022-03-16 09:43:34
## 3 EFE527AF80B66109  classic_bike 2022-03-23 19:52:02 2022-03-23 19:54:48
## 4 9F446FD9DEE3F389  classic_bike 2022-03-01 19:12:26 2022-03-01 19:22:14
## 5 431128AD9AFFEDC0  classic_bike 2022-03-21 18:37:01 2022-03-21 19:19:11
##                   start_station_name start_station_id
## 1             Wabash Ave & Wacker Pl     TA1307000131
## 2              Michigan Ave & Oak St            13042
## 3              Broadway & Berwyn Ave            13109
## 4             Wabash Ave & Wacker Pl     TA1307000131
## 5 DuSable Lake Shore Dr & North Blvd           LF-005
##                       end_station_name end_station_id start_lat start_lng
## 1             Kingsbury St & Kinzie St   KA1503000043  41.88688 -87.62603
## 2 Orleans St & Chestnut St (NEXT Apts)            620  41.90100 -87.62375
## 3                 Broadway & Ridge Ave          15578  41.97835 -87.65975
## 4           Franklin St & Jackson Blvd   TA1305000025  41.88688 -87.62603
## 5             Loomis St & Jackson Blvd          13206  41.91172 -87.62680
##    end_lat   end_lng member_casual
```

```
## 1 41.88918 -87.63851        member
## 2 41.89820 -87.63754        member
## 3 41.98404 -87.66027        member
## 4 41.87771 -87.63532        member
## 5 41.87794 -87.66201        member
```

```
head(apr22, n = 5)
```

```
##            ride_id rideable_type          started_at            ended_at
## 1 3564070EEFD12711 electric_bike 2022-04-06 17:42:48 2022-04-06 17:54:36
## 2 0B820C7FCF22F489  classic_bike 2022-04-24 19:23:07 2022-04-24 19:43:17
## 3 89EEEE32293F07FF  classic_bike 2022-04-20 19:29:08 2022-04-20 19:35:16
## 4 84D4751AEB31888D  classic_bike 2022-04-22 21:14:06 2022-04-22 21:23:29
## 5 5664BCF0D1DE7A8B electric_bike 2022-04-16 15:56:30 2022-04-16 16:02:11
##          start_station_name start_station_id         end_station_name
## 1    Paulina St & Howard St              515  University Library (NU)
## 2 Wentworth Ave & Cermak Rd            13075    Green St & Madison St
## 3      Halsted St & Polk St     TA1307000121    Green St & Madison St
## 4 Wentworth Ave & Cermak Rd            13075 Delano Ct & Roosevelt Rd
## 5      Halsted St & Polk St     TA1307000121  Clinton St & Madison St
##   end_station_id start_lat start_lng  end_lat   end_lng member_casual
## 1            605  42.01913 -87.67353 42.05294 -87.67345        member
## 2   TA1307000120  41.85308 -87.63193 41.88189 -87.64879        member
## 3   TA1307000120  41.87184 -87.64664 41.88189 -87.64879        member
## 4   KA1706005007  41.85308 -87.63193 41.86749 -87.63219        casual
## 5   TA1305000032  41.87181 -87.64657 41.88224 -87.64107        member
```

```
head(may22, n = 5)
```

```
##            ride_id rideable_type          started_at            ended_at
## 1 EC2DE40644C6B0F4  classic_bike 2022-05-23 23:06:58 2022-05-23 23:40:19
## 2 1C31AD03897EE385  classic_bike 2022-05-11 08:53:28 2022-05-11 09:31:22
## 3 1542FBEC830415CF  classic_bike 2022-05-26 18:36:28 2022-05-26 18:58:18
## 4 6FF59852924528F8  classic_bike 2022-05-10 07:30:07 2022-05-10 07:38:49
## 5 483C52CAAE12E3AC  classic_bike 2022-05-10 17:31:56 2022-05-10 17:36:57
##                  start_station_name start_station_id
## 1            Wabash Ave & Grand Ave     TA1307000117
## 2 DuSable Lake Shore Dr & Monroe St            13300
## 3           Clinton St & Madison St     TA1305000032
## 4           Clinton St & Madison St     TA1305000032
## 5           Clinton St & Madison St     TA1305000032
##              end_station_name end_station_id start_lat start_lng  end_lat
## 1      Halsted St & Roscoe St   TA1309000025  41.89147 -87.62676 41.94367
## 2 Field Blvd & South Water St          15534  41.88096 -87.61674 41.88635
## 3     Wood St & Milwaukee Ave          13221  41.88224 -87.64107 41.90765
## 4      Clark St & Randolph St   TA1305000030  41.88224 -87.64107 41.88458
## 5         Morgan St & Lake St   TA1306000015  41.88224 -87.64107 41.88578
##     end_lng member_casual
## 1 -87.64895        member
## 2 -87.61752        member
## 3 -87.67255        member
## 4 -87.63189        member
## 5 -87.65102        member
```

```
head(jun22, n = 5)
```

```
##            ride_id rideable_type          started_at            ended_at
## 1 600CFD130D0FD2A4 electric_bike 2022-06-30 17:27:53 2022-06-30 17:35:15
## 2 F5E6B5C1682C6464 electric_bike 2022-06-30 18:39:52 2022-06-30 18:47:28
## 3 B6EB6D27BAD771D2 electric_bike 2022-06-30 11:49:25 2022-06-30 12:02:54
## 4 C9C320375DE1D5C6 electric_bike 2022-06-30 11:15:25 2022-06-30 11:19:43
## 5 56C055851023BE98 electric_bike 2022-06-29 23:36:50 2022-06-29 23:45:17
##   start_station_name start_station_id end_station_name end_station_id start_lat
## 1                                                                        41.89
## 2                                                                        41.91
## 3                                                                        41.91
## 4                                                                        41.80
## 5                                                                        41.91
##   start_lng end_lat end_lng member_casual
## 1    -87.62   41.91  -87.62        casual
## 2    -87.62   41.93  -87.63        casual
## 3    -87.65   41.89  -87.61        casual
## 4    -87.66   41.80  -87.65        casual
## 5    -87.63   41.93  -87.64        casual
```

Binding all the dataframes together into a single dataframe

```
all_data <- bind_rows(jul21,aug21,sep21,oct21,nov21,dec21,jan22,feb22,mar22,apr22,may22,jun22)
```

getting a preview of the working data

```
glimpse(all_data)
```

```
## Rows: 5,900,385
## Columns: 13
## $ ride_id            <chr> "0A1B623926EF4E16", "B2D5583A5A5E76EE", "6F264597DD~
## $ rideable_type      <chr> "docked_bike", "classic_bike", "classic_bike", "cla~
## $ started_at         <chr> "2021-07-02 14:44:36", "2021-07-07 16:57:42", "2021~
## $ ended_at           <chr> "2021-07-02 15:19:58", "2021-07-07 17:16:09", "2021~
## $ start_station_name <chr> "Michigan Ave & Washington St", "California Ave & C~
## $ start_station_id   <chr> "13001", "17660", "SL-012", "17660", "17660", "1766~
## $ end_station_name   <chr> "Halsted St & North Branch St", "Wood St & Hubbard ~
## $ end_station_id     <chr> "KA1504000117", "13432", "KA1503000044", "13196", "~
## $ start_lat          <dbl> 41.88398, 41.90036, 41.86038, 41.90036, 41.90035, 4~
## $ start_lng          <dbl> -87.62468, -87.69670, -87.62581, -87.69670, -87.696~
## $ end_lat            <dbl> 41.89937, 41.88990, 41.89017, 41.89456, 41.88659, 4~
## $ end_lng            <dbl> -87.64848, -87.67147, -87.62619, -87.65345, -87.658~
## $ member_casual      <chr> "casual", "casual", "member", "member", "casual", "~
```

Data Credibility and Bias

This is an **O**riginal data from a **R**eliable organization and is **C**omprehensive, **C**urrent and **C**ited. Since the data set contains data from the entire population, sampling bias has been evaded. Since usage of personally identifiable information has been prohibited, it is impossible to determine daily riders and frequency of usage of a particular casual rider over the course of a week or a month.

Process Phase

I will do my analysis in R due to the accessibility, the amount of data I have to handle and to be able to create data visualization to share my results with stakeholders.

I will be using the following libraries - ggplot2 - tidyverse - lubridate - dplyr - skmir - janitor

```
library(ggplot2)
library(tidyverse)
library(lubridate)
library(dplyr)
library(skimr)
library(janitor)
```

Cleaning

```
#Rearranging the data frame by starting time (started_at) and storing as variable
clean_data <- arrange(all_data, started_at)

#renaming columns for accessibility
clean_data <- rename(clean_data, trip_id = ride_id, bike = rideable_type,
                     startplace = start_station_name, endplace = end_station_name,
                     start_id = start_station_id, end_id = end_station_id,
                     membership = member_casual)

#dropping NA values
clean_data <- drop_na(clean_data)

#checking number of unique
n_unique(clean_data$ride_id)
```

```
## [1] 0
```

```
#separating date, month, day, starting hour and calculate day of the week
clean_data$date <- as.Date(clean_data$started_at)
clean_data$month <- month(clean_data$date, label = TRUE)
clean_data$day <- day(clean_data$date)
clean_data$weekday <- wday(clean_data$date, label = TRUE, abbr = FALSE)
clean_data$start_hour <- format(as_datetime(clean_data$started_at), "%H")

#making a separate row for year and month
clean_data$year_month <- format(as.Date(clean_data$date), "%Y-%m")

#calculating ride length
clean_data$ride_length <- difftime(clean_data$ended_at, clean_data$started_at)

#finding ride lengths that are negative, if any
neg_length <- filter(clean_data, ride_length < 0)

#removing bad data
clean_data <- subset(clean_data, ride_length >= 0)
```

```r
#to ensure numerical data isn't shown in scientific format
options(scipen = 10)
```

Analyze Phase

Summary of the dataset

```r
summary(clean_data)
```

```
##    trip_id              bike            started_at          ended_at
##  Length:5894865     Length:5894865     Length:5894865     Length:5894865
##  Class :character   Class :character   Class :character   Class :character
##  Mode  :character   Mode  :character   Mode  :character   Mode  :character
##
##
##
##
##   startplace          start_id           endplace            end_id
##  Length:5894865     Length:5894865     Length:5894865     Length:5894865
##  Class :character   Class :character   Class :character   Class :character
##  Mode  :character   Mode  :character   Mode  :character   Mode  :character
##
##
##
##
##    start_lat       start_lng         end_lat         end_lng
##  Min.   :41.64   Min.   :-87.84   Min.   :41.39   Min.   :-88.97
##  1st Qu.:41.88   1st Qu.:-87.66   1st Qu.:41.88   1st Qu.:-87.66
##  Median :41.90   Median :-87.64   Median :41.90   Median :-87.64
##  Mean   :41.90   Mean   :-87.65   Mean   :41.90   Mean   :-87.65
##  3rd Qu.:41.93   3rd Qu.:-87.63   3rd Qu.:41.93   3rd Qu.:-87.63
##  Max.   :45.64   Max.   :-73.80   Max.   :42.17   Max.   :-87.49
##
##   membership             date                month             day
##  Length:5894865     Min.   :2021-07-01   Jul    : 821666   Min.   : 1.00
##  Class :character   1st Qu.:2021-08-26   Aug    : 803617   1st Qu.: 8.00
##  Mode  :character   Median :2021-10-27   Jun    : 768137   Median :16.00
##                     Mean   :2021-12-11   Sep    : 755516   Mean   :15.69
##                     3rd Qu.:2022-04-25   May    : 634135   3rd Qu.:23.00
##                     Max.   :2022-06-30   Oct    : 630742   Max.   :31.00
##                                          (Other):1481052
##       weekday        start_hour         year_month        ride_length
##  Sunday   :864968   Length:5894865     Length:5894865     Length:5894865
##  Monday   :772371   Class :character   Class :character   Class :difftime
##  Tuesday  :795413   Mode  :character   Mode  :character   Mode  :numeric
##  Wednesday:801480
##  Thursday :850391
##  Friday   :831607
##  Saturday :978635
```

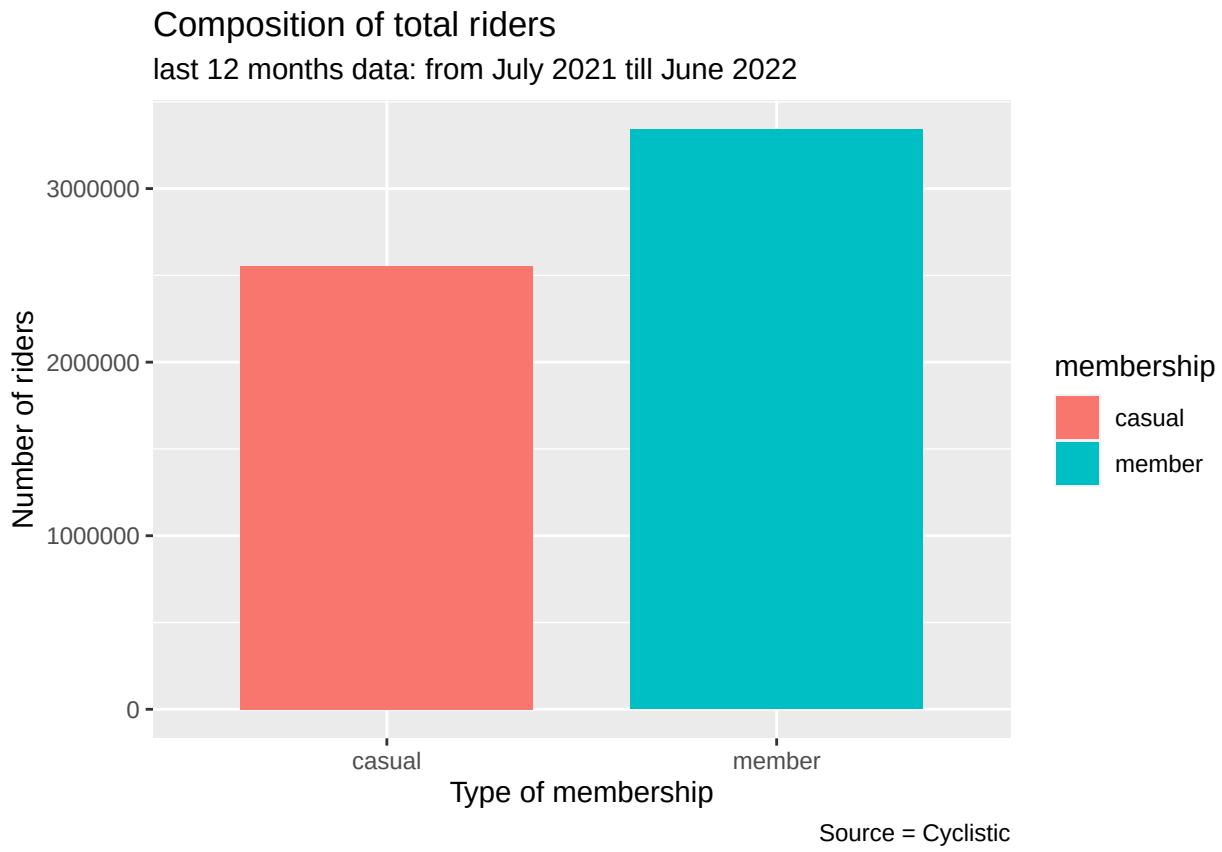
The Analysis

Composition of Riders

```
#composition of total riders
clean_data %>%
   group_by(membership) %>%
   summarise(riders = n())
```

```
## # A tibble: 2 x 2
##   membership  riders
##   <chr>        <int>
## 1 casual     2553655
## 2 member     3341210
```

```
ggplot(data = clean_data) +
   geom_bar(mapping = aes(x = membership, fill = membership), width = 0.75) +
   labs(title = "Composition of total riders",
        subtitle = "last 12 months data: from July 2021 till June 2022",
        caption = "Source = Cyclistic",
        x = "Type of membership", y = "Number of riders")
```

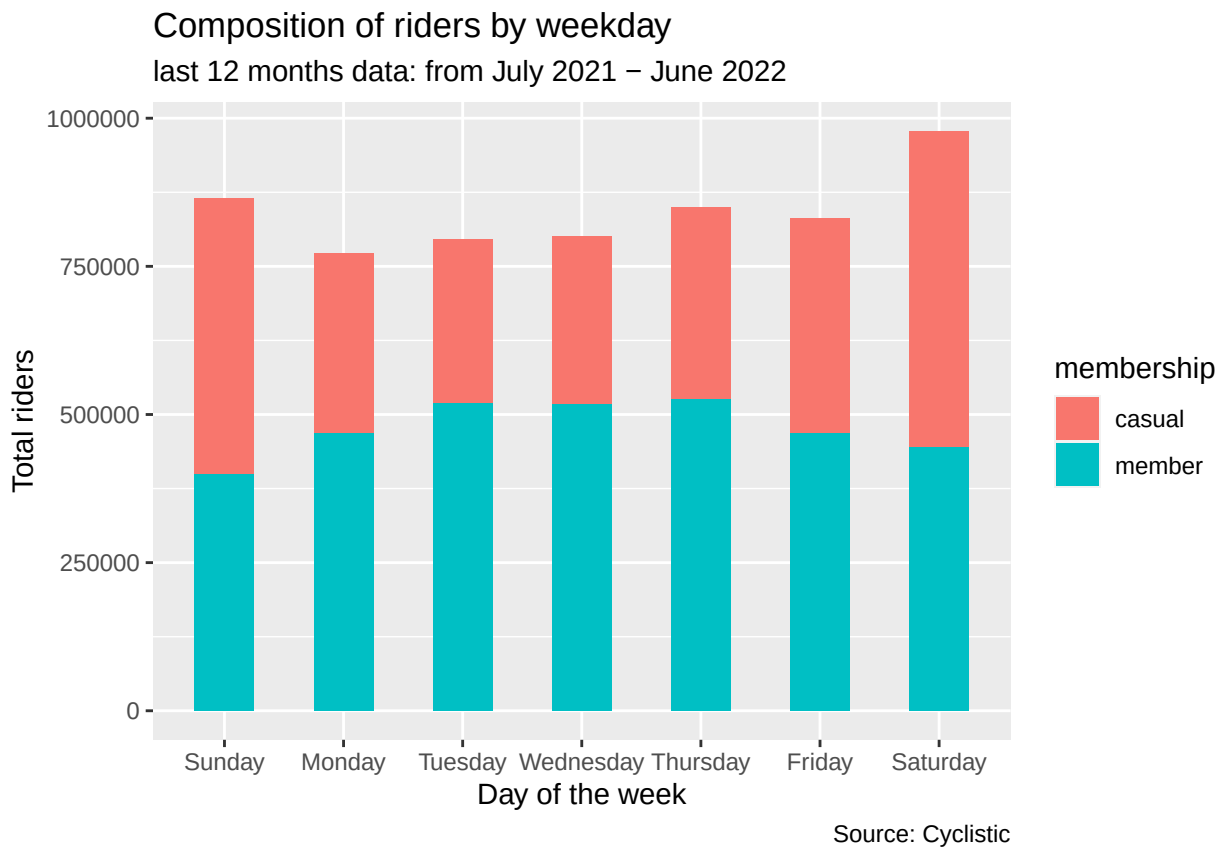


```r
#composition of riders by weekday
clean_data %>%
   group_by(weekday, membership) %>%
   summarise(riders = n())
```

```
## `summarise()` has grouped output by 'weekday'. You can override using the
## `.groups` argument.
```

```
## # A tibble: 14 x 3
## # Groups:   weekday [7]
##    weekday   membership riders
##    <ord>     <chr>       <int>
##  1 Sunday    casual     466178
##  2 Sunday    member     398790
##  3 Monday    casual     303454
##  4 Monday    member     468917
##  5 Tuesday   casual     277297
##  6 Tuesday   member     518116
##  7 Wednesday casual     285031
##  8 Wednesday member     516449
##  9 Thursday  casual     324596
## 10 Thursday  member     525795
## 11 Friday    casual     362482
## 12 Friday    member     469125
## 13 Saturday  casual     534617
## 14 Saturday  member     444018
```

```r
ggplot(data = clean_data) +
   geom_bar(mapping = aes(x = weekday, fill = membership), width = 0.5) +
   labs(title = "Composition of riders by weekday",
        subtitle = "last 12 months data: from July 2021 - June 2022",
        caption = "Source: Cyclistic",
        x = "Day of the week", y = "Total riders")
```

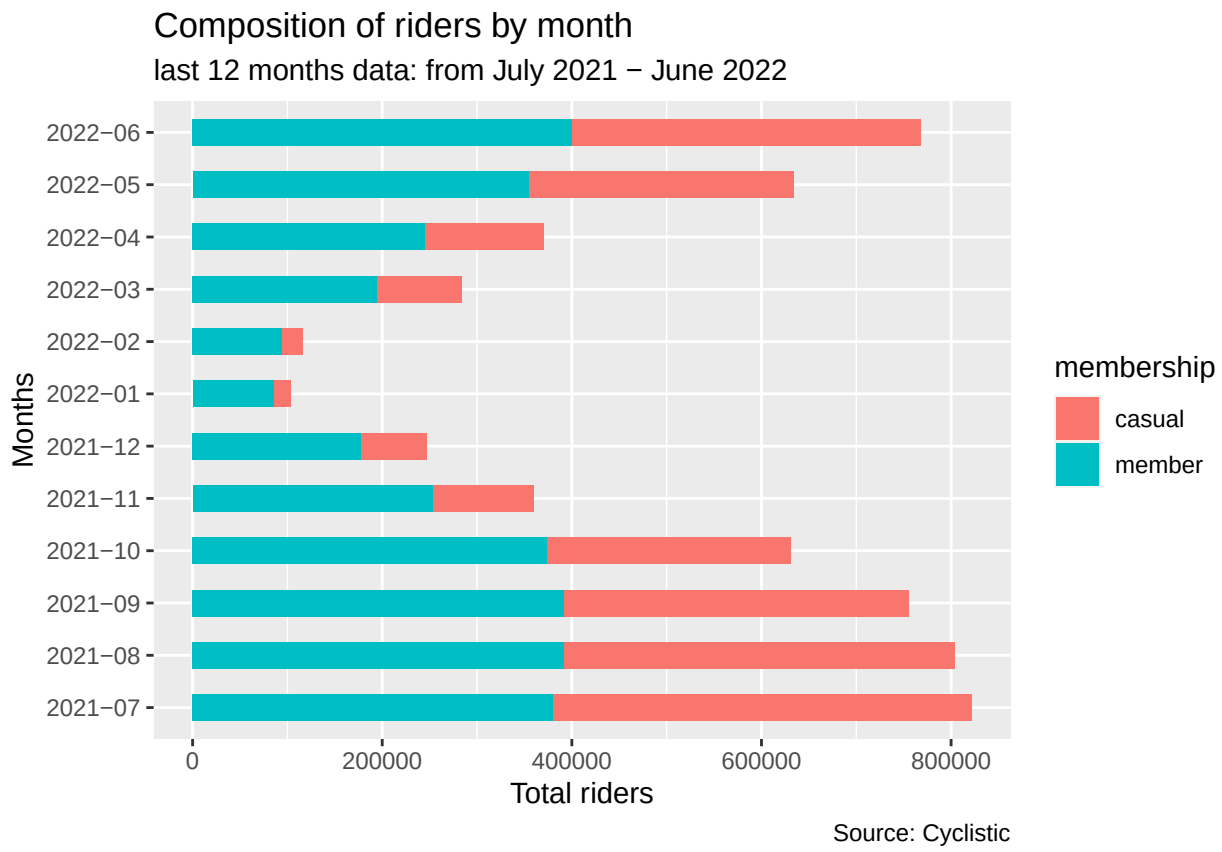



```
#composition of riders by month
clean_data %>%
   group_by(year_month, membership) %>%
   summarise(riders = n())
```

```
## `summarise()` has grouped output by 'year_month'. You can override using the
## `.groups` argument.
```

```
## # A tibble: 24 x 3
## # Groups:   year_month [12]
##    year_month membership riders
##    <chr>      <chr>       <int>
##  1 2021-07    casual     441465
##  2 2021-07    member     380201
##  3 2021-08    casual     412101
##  4 2021-08    member     391516
##  5 2021-09    casual     363460
##  6 2021-09    member     392056
##  7 2021-10    casual     256826
##  8 2021-10    member     373916
##  9 2021-11    casual     106755
## 10 2021-11    member     252979
## # ... with 14 more rows
```

```
ggplot(data = clean_data) +
   geom_bar(mapping = aes(y = year_month, fill = membership), width = 0.5) +
   labs(title = "Composition of riders by month",
        subtitle = "last 12 months data: from July 2021 - June 2022",
        caption = "Source: Cyclistic",
        y = "Months", x = "Total riders")
```



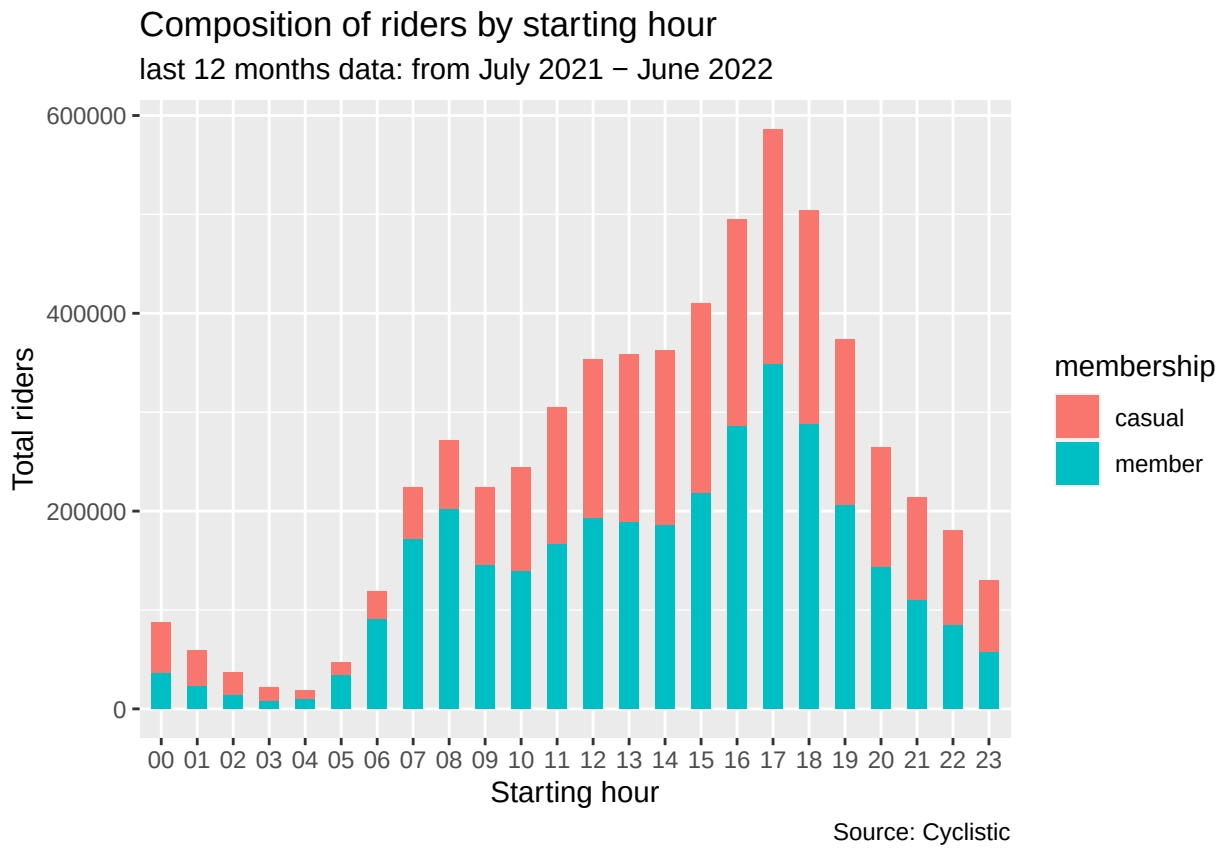

```
#composition of riders by hour of day
clean_data %>%
   group_by(start_hour, membership) %>%
   summarise(riders = n())
```

```
## `summarise()` has grouped output by 'start_hour'. You can override using the
## `.groups` argument.
```

```
## # A tibble: 48 x 3
## # Groups:   start_hour [24]
##    start_hour membership riders
##    <chr>      <chr>       <int>
##  1 00         casual      52071
##  2 00         member      35613
##  3 01         casual      36387
##  4 01         member      22972
##  5 02         casual      24077
```

```
##  6 02          member      13356
##  7 03          casual      13674
##  8 03          member       7926
##  9 04          casual       9707
## 10 04          member       9173
## # ... with 38 more rows
```

```
ggplot(data = clean_data) +
   geom_bar(mapping = aes(x = start_hour, fill = membership), width = 0.55) +
   labs(title = "Composition of riders by starting hour",
        subtitle = "last 12 months data: from July 2021 - June 2022",
        caption = "Source: Cyclistic",
        x = "Starting hour", y = "Total riders")
```



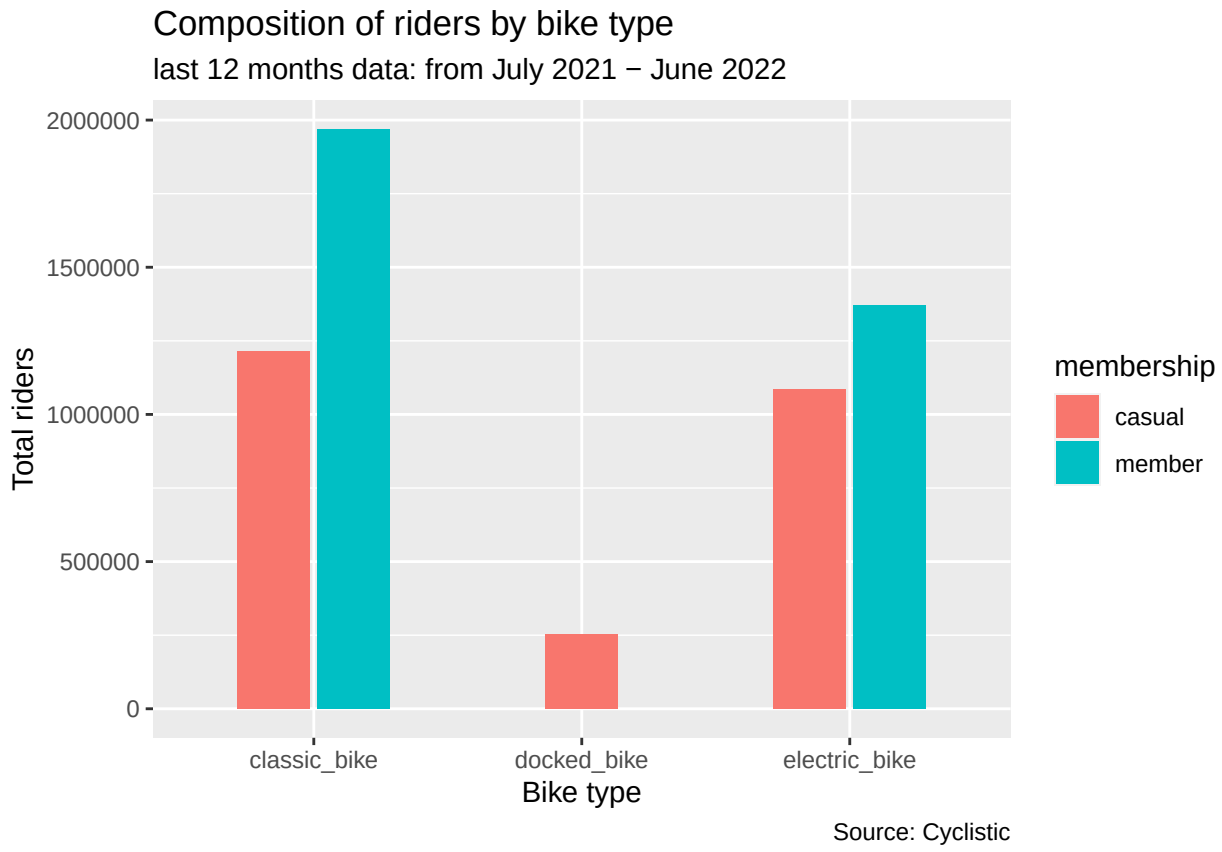

```
#composition of riders by bike type
clean_data %>%
   group_by(bike, membership) %>%
   summarise(riders = n())
```

```
## `summarise()` has grouped output by 'bike'. You can override using the
## `.groups` argument.
```

```
## # A tibble: 5 x 3
## # Groups:   bike [3]
##   bike          membership  riders
```

```
##   <chr>         <chr>        <int>
## 1 classic_bike  casual     1215072
## 2 classic_bike  member     1970173
## 3 docked_bike   casual      252048
## 4 electric_bike casual     1086535
## 5 electric_bike member     1371037
```

```
ggplot(data = clean_data) +
   geom_bar(mapping = aes(x = bike, fill = membership), width = 0.60,
            position = position_dodge2(preserve = "single")) +
   labs(title = "Composition of riders by bike type",
        subtitle = "last 12 months data: from July 2021 - June 2022",
        caption = "Source: Cyclistic",
        x = "Bike type", y = "Total riders")
```



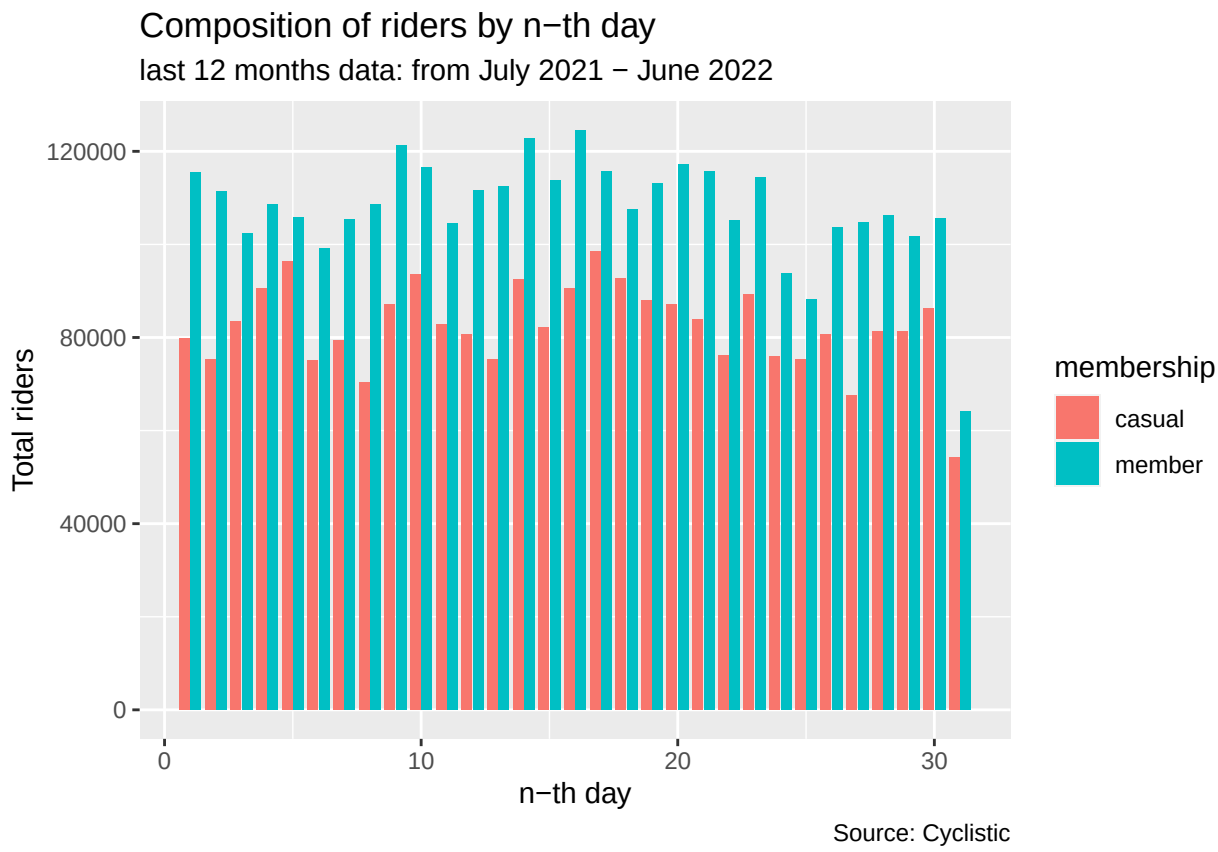

```
#composition of riders by day
clean_data %>%
   group_by(day, membership) %>%
   summarise(riders = n())
```

```
## `summarise()` has grouped output by 'day'. You can override using the `.groups`
## argument.
```

```
## # A tibble: 62 x 3
## # Groups:   day [31]
```

```
##       day membership riders
##     <int> <chr>       <int>
##  1      1 casual      79858
##  2      1 member     115538
##  3      2 casual      75370
##  4      2 member     111301
##  5      3 casual      83558
##  6      3 member     102356
##  7      4 casual      90605
##  8      4 member     108703
##  9      5 casual      96461
## 10      5 member     105798
## # ... with 52 more rows
```

```
ggplot(data = clean_data) +
   geom_bar(mapping = aes(x = day, fill = membership), width = 0.9,
            position = position_dodge2(preserve = "single")) +
   labs(title = "Composition of riders by n-th day",
        subtitle = "last 12 months data: from July 2021 - June 2022",
        caption = "Source: Cyclistic",
        x = "n-th day", y = "Total riders")
```



Composition of riders by Riding time

```
#saving a separate data frame for calculations regarding riding time
time <- clean_data %>%
   group_by(membership) %>%
   select(year_month, weekday, day, start_hour, membership, bike, ride_length)

#separating out the ride time for casual riders
time_casual <- time %>%
   filter(membership == "casual") %>%
   select(year_month, weekday, day, start_hour, membership, bike, ride_length)

#calculating outliers
lower_bound <- quantile(time_casual$ride_length, 0.025)
lower_bound
```

```
## Time difference of 124 secs
```

```
upper_bound <- quantile(time_casual$ride_length, 0.975)
upper_bound
```

```
## Time difference of 6284 secs
```

```
#eliminating the outliers and saving in a different vaariable
time_casual <- time_casual[time_casual$ride_length > lower_bound &
                           time_casual$ride_length < upper_bound, ]

#separating out the ride time for members
time_members <- time %>%
   filter(membership == "member") %>%
   select(year_month, weekday, day, start_hour, membership, bike, ride_length)

#calculating outliers
lower_bound <- quantile(time_members$ride_length, 0.025)
lower_bound
```

```
## Time difference of 89 secs
```

```
upper_bound <- quantile(time_members$ride_length, 0.975)
upper_bound
```

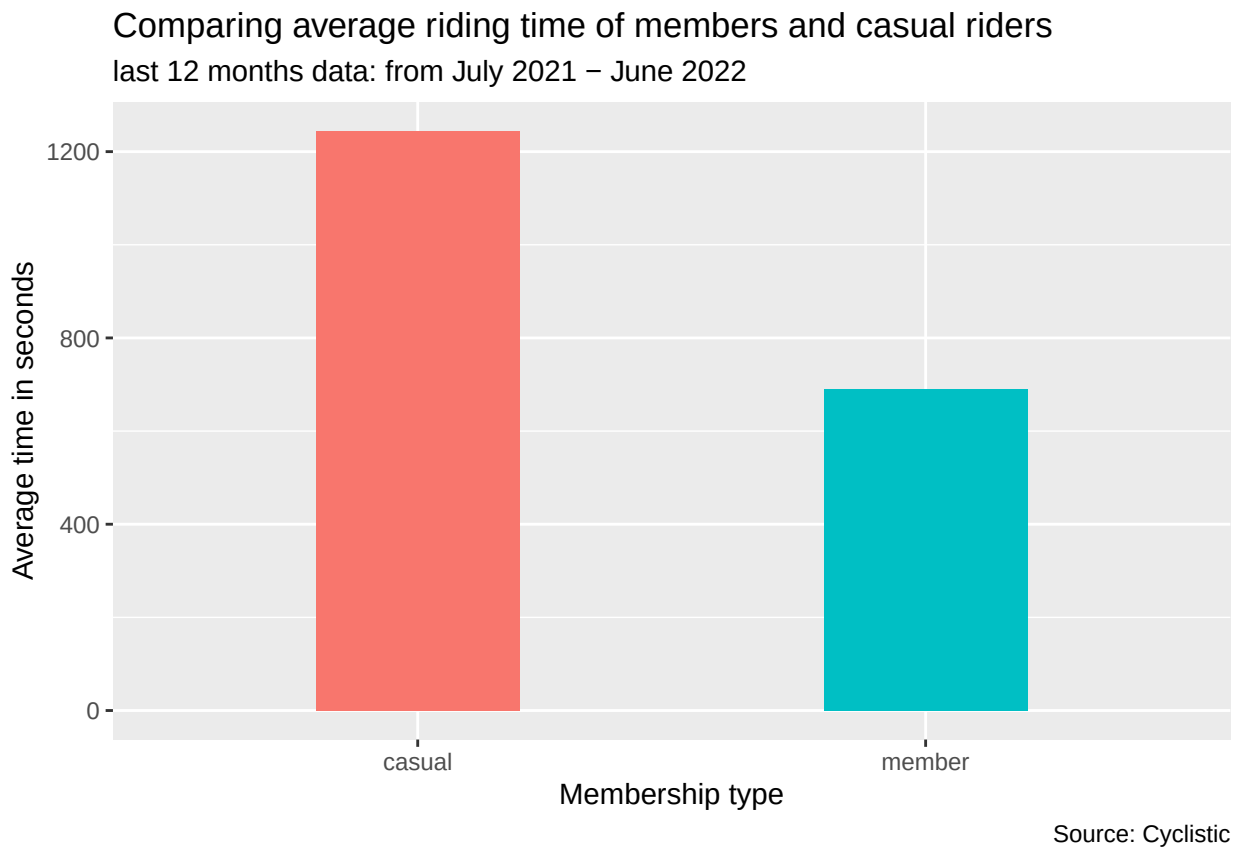
```
## Time difference of 2454 secs
```

```
#eliminating the outliers
time_members <- time_members[time_members$ride_length > lower_bound &
                             time_members$ride_length < upper_bound, ]

#combining the time data
time <- bind_rows(time_casual, time_members)
```

```
#finding average riding time by membership
time %>%
   group_by(membership) %>%
   summarise(avg_time = mean(ride_length)) %>%
   ggplot(aes(x = membership, y = avg_time, fill = membership))+
   geom_col(width = 0.4, show.legend = FALSE)+
   labs(title = "Comparing average riding time of members and casual riders",
        subtitle = "last 12 months data: from July 2021 - June 2022",
        caption = "Source: Cyclistic",
        x = "Membership type", y = "Average time in seconds")
```

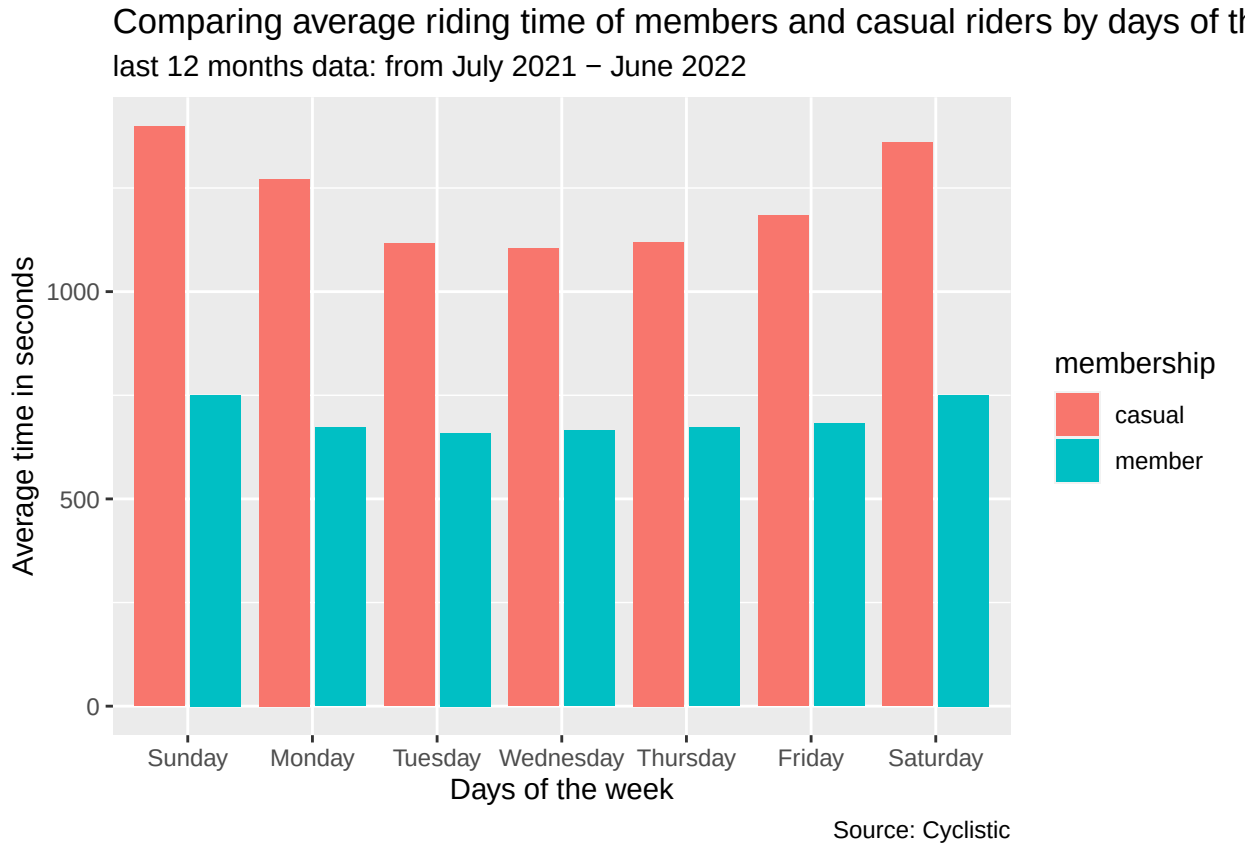
```
## Don't know how to automatically pick scale for object of type difftime. Defaulting to continuous.
```




```
#finding average riding time by membership by weekday
time %>%
   group_by(weekday, membership) %>%
   summarise(avg_time = mean(ride_length)) %>%
   ggplot(aes(x = weekday, y = avg_time, fill = membership))+
   geom_col(position = "dodge2")+
   labs(title = "Comparing average riding time of members and casual riders by days of the week",
        subtitle = "last 12 months data: from July 2021 - June 2022",
        caption = "Source: Cyclistic",
        x = "Days of the week", y = "Average time in seconds")
```

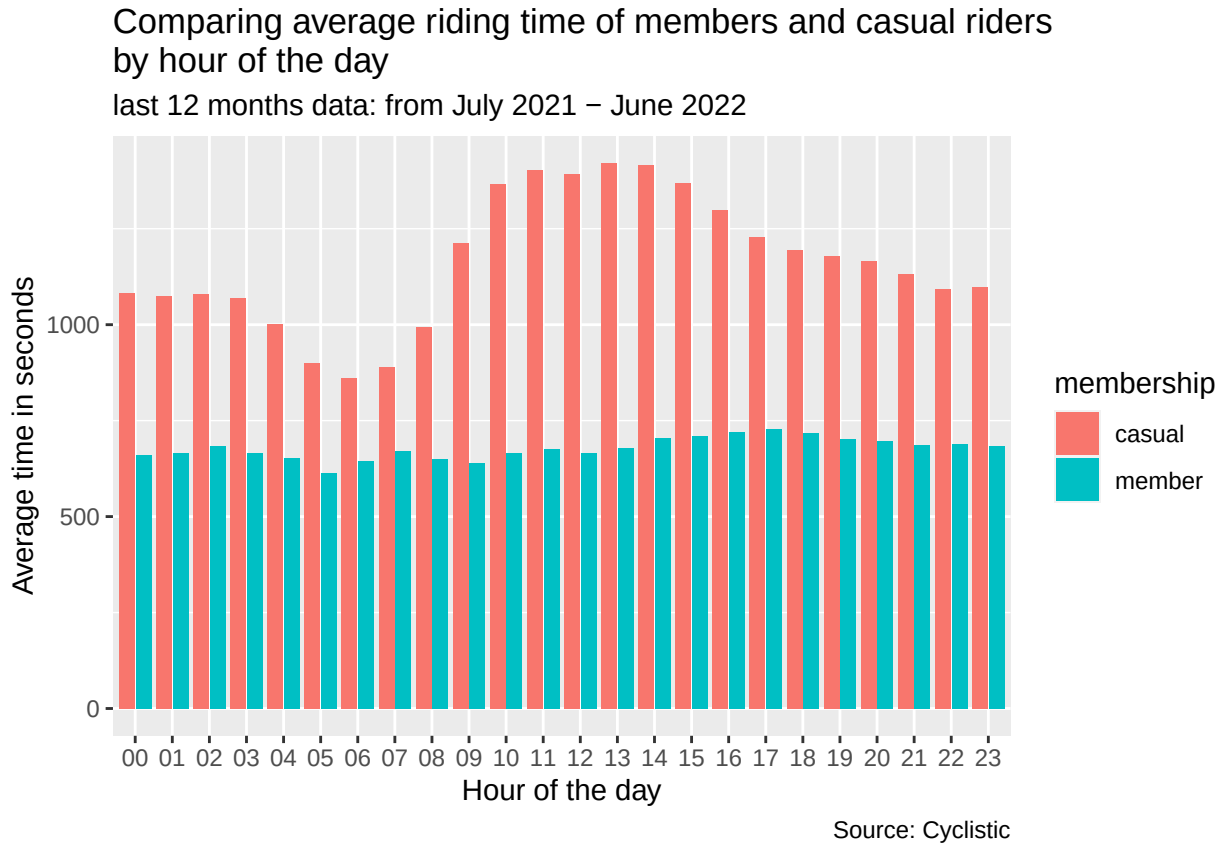
```
## `summarise()` has grouped output by 'weekday'. You can override using the
```

```
## `.groups` argument.
## Don't know how to automatically pick scale for object of type difftime.
## Defaulting to continuous.
```



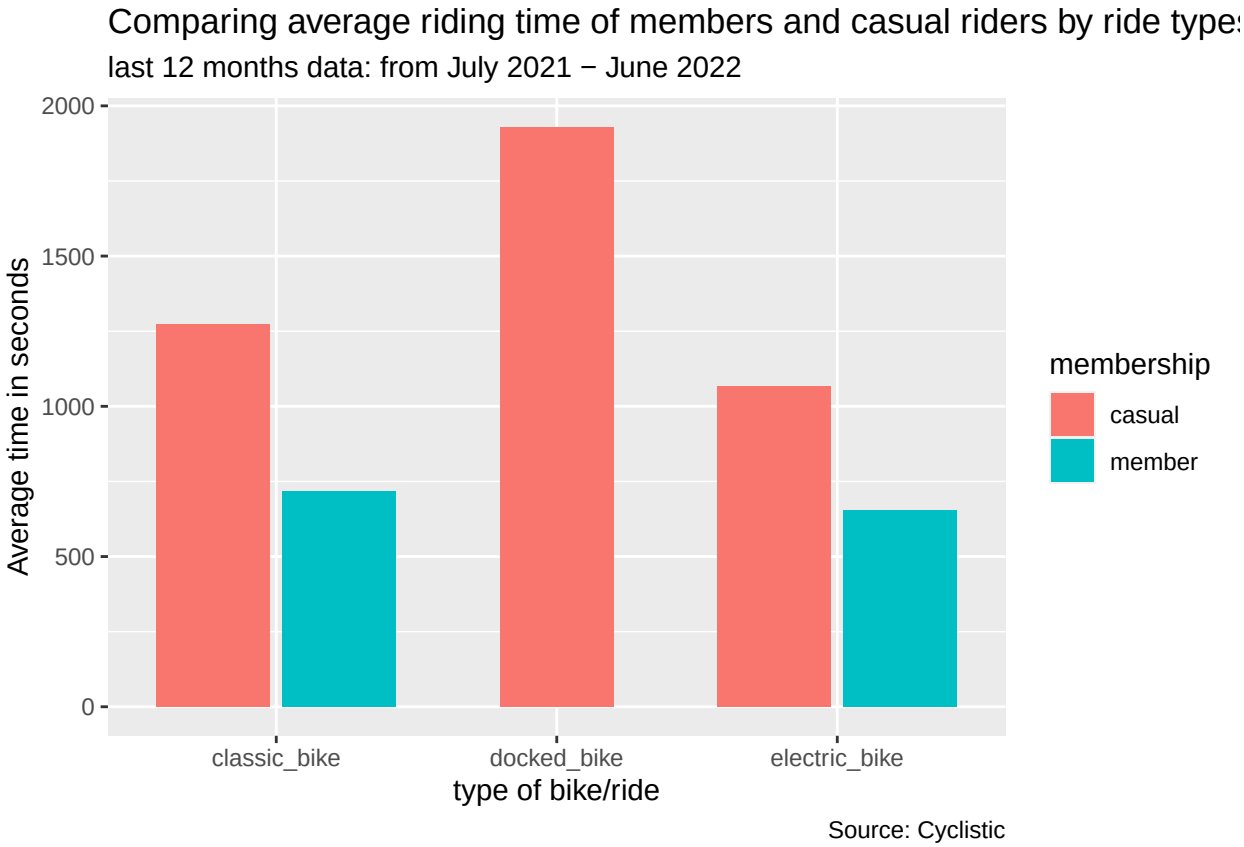

```
#finding average riding time by membership by starting hour
time %>%
   group_by(start_hour, membership) %>%
   summarise(avg_time = mean(ride_length)) %>%
   ggplot(aes(x = start_hour, y = avg_time, fill = membership))+
   geom_col(position = "dodge2")+
   labs(title = "Comparing average riding time of members and casual riders
by hour of the day",
        subtitle = "last 12 months data: from July 2021 - June 2022",
        caption = "Source: Cyclistic",
        x = "Hour of the day", y = "Average time in seconds")
```

```
## `summarise()` has grouped output by 'start_hour'. You can override using the
## `.groups` argument.
## Don't know how to automatically pick scale for object of type difftime.
## Defaulting to continuous.
```




```
#bike type
time %>%
   group_by(bike, membership) %>%
   summarise(avg_time = mean(ride_length)) %>%
   ggplot(aes(x = bike, y = avg_time, fill = membership))+
   geom_col(position = position_dodge2(preserve = "single"))+
   labs(title = "Comparing average riding time of members and casual riders by ride types",
        subtitle = "last 12 months data: from July 2021 - June 2022",
        caption = "Source: Cyclistic",
        x = "type of bike/ride", y = "Average time in seconds")
```

```
## `summarise()` has grouped output by 'bike'. You can override using the
## `.groups` argument.
## Don't know how to automatically pick scale for object of type difftime.
## Defaulting to continuous.
```



Key findings

1. More members use Cyclistic than casual riders.
2. Observation as per days of the week -
    - Weekends see more riders than weekdays
    - Members avail more rides during the midweek than the weekends
    - Casual riders avail more rides during the weekends, surpassing the member riders in those two days
3. Observation on a monthly basis -
    - Warm months have more riders than cold months
    - Total ridership peaks in mid year
    - Ridership varies greatly among casual riders in the warm and cold months
    - Significant portion of the riders in cold months are members
    - Mid-year ridership of casuals and members are comparable
4. According to bike types, or ride types -
    - People using docked bikes are very small compared to other ride types
    - Preference for classic bikes and electric bikes is close for casual riders
    - Members prefer classic bikes than electric bikes
5. Distribution of data according to the particular date of the month -
    - 31^{st} has the minimum value for both casual and member riders since only few months have 31^{st} day
    - Ridership for members peak at the middle of the month

- First half of the month sees more members than the second half
- Ridership for casuals increases till the middle of the first week and drops significantly at the end of the week, rising to a 'local' peak at the 10^{th} of the month and again drops at the end of the week, followed by a rise on 17^{th} and drops gradually for the rest of the month except for a sudden rise at the end of the month
- Fluctuation in ridership is more in the first half than in the second half

6. Casual riders spend more time in riding than members

    - casuals spend more times in the weekends
    - casuals spend least time in mid-week
    - members' riding time remain comparable throughout
    - riding time of members remain comparable throughout the day
    - riding time of casuals peak at noon and valleys in the morning
    - casuals spend more time on classic bike than electric bike
    - members spend almost equal time on classic bike and electric bike

Share Phase

I have decided to share my analysis as a slide which can be found here

Act Phase

Based on the analysis of the riders' data for the past 12 months, from July 2022 to June 2021, I am sharing some recommendations for future success of Cyclistic -

1. **Weekend membership plan**: Since we have seen that the casual riders travel more in weekends, this type of membership will attract at least those casual riders who avail Cyclistic services every weekend.
2. **Bi-annual membership plan**: Since we see that casual riders avail the services more in the warm months, this type of membership will attract casual riders into becoming a member for half a year.
3. **Discounts on annual membership based on riding time**: This would also attract casual riders since we have seen that casual riders travel for a longer duration than members.